

Level: 1st Year
Material: Algorithms and Static Data Structures

General guidelines

- Each exercise must be written on a separate sheet.
- Documents are strictly prohibited.
- Smartphones, smartwatches and calculators are strictly prohibited.

Exercise 01: (07 points)

An integer number n is considered as valid if each digit appears at most twice, no identical digits are adjacent, and the sum of every block of three consecutive digits is odd.

- Develop the Procedure `numberToArray(n: integer; VAR digitArray: TDigit; VAR length: integer)` that converts the number n into an array of digits `digitArray` of a size `length`. (1.5 points)
- Develop the Function `checkDigits(digitArray: TDigit; sizeDigit: integer): Boolean` to check if each digit of the array `digitArray` of size `sizeDigit` appears at most twice, and if two identical digits are adjacent. (1.5 points)
- Develop the Function `checkTripletSum(digitArray: TDigit; sizeDigit: integer): Boolean` that checks if the sum of every block of three consecutive digits of the array `digitArray` is odd. (1.5 points)
- Develop the Function `validNumber(n: Integer): Boolean` that invokes the previous functions and checks whether the number n is valid or not. (1.5 points)
- Write the main algorithm while defining the different types: `TDigit` for `digitArray` and `TAppearence` for `appearArray`. (1.0 point)

Example:

$n=251463$, `digitArray=`

2	5	1	4	6	3
---	---	---	---	---	---

, number $n=251463$ is valid.

$n=112345$, `digitArray=`

1	1	2	3	4	5
---	---	---	---	---	---

, number $n=112345$ is invalid (digit 1 repeated adjacent).

NB:

- For the Function `checkDigits`, consider another array `appearArray` to save the frequency of appearance of each digit in the array `digitArray`.

Exercise 02: (6.5 points)

Cocktail sort is a bidirectional bubble sort: (i) The forward pass traverses the array from the left to the right comparing and swapping adjacent elements. This pass moves the largest element to the right. (ii) The backward pass traverses the array from the right to the left and does the same. This pass moves the smallest element to the left.

- Write the Procedure `exchangeXY(VAR x, y: Integer)` that exchanges two integers x and y . (0.5 point)
- Write the Procedure `forwardPass(VAR Tab: TArray; VAR start, end: Integer; VAR swapped: Boolean)` that implements the forward pass. (1.5 points)
- Write the Procedure `backwardPass(VAR Tab: TArray; VAR start, end: Integer; VAR swapped: Boolean)` that implements the backward pass. (1.5 points)
- Write the Procedure `cocktailSort(VAR Tab: TArray; size: integer)` that implements the cocktail sort by invoking the previous procedures. (2.0 points)
- Write the main algorithm invoking the previous modules, while defining the related type `TArray`. (1.0 point)

NB:

- Use the procedure `readArray(VAR Tab: TArray; size: integer)` that reads the values of the array.
- Use the procedure `printArray (Tab: TArray; size: integer)` that prints the values of the array.

First semester final exam

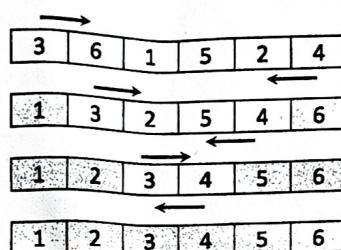
Level: 1st Year

Material: Algorithms and Static Data Structures

Date: 15/01/2026

Duration: 02h00

Example:



Exercise 03: (6.5 points)

Given a string `inputStr` containing several words separated by single spaces. Write an algorithm that reverses each word individually while keeping the order of the words unchanged.

- Write the Function `reverseWord(word: string): string` that reverses the input word. (2.0 points)
- Write the Procedure `extractWord(inputStr: string; startIndex: integer; VAR reversedWord: string; VAR newIndex: integer)` that extracts the current word of the sentence and reverses it by invoking the previous function `reverseWord`. (2.0 points)
- Develop the Function `reverseAllWord(inputStr: string): string` that processes all the words of the input sentence. (2.0 points)
- Write the main algorithm. (0.5 point)

Example

- `inputStr = "ALGORITHMS AND STATIC DATA STRUCTURES".`
- `reversedStr = "SMHTIROGLA DNA CITATS ATAD SERUTCURTS".`

NB:

- Use the function `getLength(inputStr: String): integer`.

Good Luck